

**Федеральное агентство по образованию
ГОУ ВПО «Уральский государственный
педагогический университет»**

Факультет информатики

Кафедра новых информационных технологий в образовании

**Б.Е. Стариченко
О.А. Горнов**

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по курсу «Теоретические основы информатики»

Екатеринбург 2009

УДК 371.4

ББК Ч 426.24/29

С77

РЕЦЕНЗЕНТЫ:

доктор физико-математических наук, профессор А.Н. Сесекин
кандидат педагогических наук, доцент И.Н. Слипкина

С77 Стариченко Б.Е., Горнов О.А.

Лабораторный практикум по курсу «Теоретические основы информатики»/ Урал. гос. пед. ун-т. Екатеринбург, 2009. – 55 с.

Пособие является частью учебно-методического комплекса для решения задач по дисциплине «Теоретические основы информатики», входящей в план подготовки студентов информационных специальностей педагогического вуза. Помимо описаний лабораторных работ, представленных в данном пособии, комплекс включает рабочие файлы, содержащие необходимые программы моделирования, а также экранные формы отчетов с генерацией индивидуальных заданий и проверкой корректности выполнения работ (в формате MS Excel); для выполнения работ требуется пакет MS Office.

УДК 371.4

ББК Ч 426.24/29

© Б.Е. Стариченко, О.А. Горнов 2009

© Уральский государственный педагогический университет 2009

Оглавление

Вводные замечания-----	4
ЛАБОРАТОРНАЯ РАБОТА 1. ЭНТРОПИЯ. ИНФОРМАЦИЯ. -----	6
ЛАБОРАТОРНАЯ РАБОТА 2. ИССЛЕДОВАНИЕ СТАТИСТИЧЕСКИХ ХАРАКТЕРИСТИК ТЕКСТА -----	11
ЛАБОРАТОРНАЯ РАБОТА 3. СРАВНЕНИЕ МЕТОДОВ КОДИРОВАНИЯ ИНФОРМАЦИИ -----	14
Часть 1. Методы алфавитного кодирования-----	15
Часть 2. Сопоставление алфавитного и блочного кодирования -	16
ЛАБОРАТОРНАЯ РАБОТА 4. КОДИРОВАНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ -----	17
Часть 1. Системы счисления-----	17
Часть 2. Нормализация чисел -----	20
Часть 3. Операции с кодами чисел-----	22
ЛАБОРАТОРНАЯ РАБОТА 5. ИССЛЕДОВАНИЕ ДИСКРЕТНЫХ ДВОИЧНЫХ КАНАЛОВ ПЕРЕДАЧИ ИНФОРМАЦИИ -----	27
Часть 1. Двоичный канал без стирания -----	28
Часть 2. Двоичный канал со стиранием -----	30
ЛАБОРАТОРНАЯ РАБОТА 6. ПОСТРОЕНИЕ ПОМЕХОУСТОЙЧИВОГО КОДА-----	31
ЛАБОРАТОРНАЯ РАБОТА 7. ИЗУЧЕНИЕ ДИСКРЕТНЫХ УСТРОЙСТВ ОБРАБОТКИ ИНФОРМАЦИИ -----	37
Часть 1. Комбинационные схемы -----	37
Часть 2. Конечные автоматы -----	41
ЛАБОРАТОРНАЯ РАБОТА 8. АЛГОРИТМИЧЕСКАЯ МАШИНА ТЬЮРИНГА-----	47

Вводные замечания

Основной целью изучения дисциплины «Теоретические основы информатики» в педагогическом вузе является освоение будущими учителями информатики базовых положений теории информации как теоретической и методологической основы других дисциплин информационно-технологической подготовки. Помимо изучения теоретических положений программа дисциплины предусматривает также знакомство с методами решения задач, связанных с представлением и обработкой дискретной информации. В настоящем пособии решение этих задач предлагается выполнять в форме компьютерных лабораторных работ.

В содержательном отношении практикум базируется на учебном пособии «Теоретические основы информатики»¹. Он включает работы по всем основным разделам курса: энтропия, информация, кодирование информации, передача информации, конечные автоматы. Перечень и последовательность выполняемых работ, а также конкретное содержание учебных заданий определяет преподаватель в зависимости от профиля специальности, а также объема учебных часов, выделенных учебным планом на данную дисциплину.

Описания лабораторных работ, приведенные в данном пособии, включают формулировку учебного задания, рекомендации по его выполнению, указания по форме отчета, перечень заданий для самостоятельной работы и контрольные вопросы. Помимо описания для выполнения заданий требуются рабочие файлы – их перечень указан в начале каждого описания; перед выполнением работы студент должен скопировать нужные файлы в свою рабочую папку.

Предполагается, что часть заданий студент выполняет самостоятельно. Отчет о работе делается по предложенной экранной форме и представляется преподавателю в виде

¹ Стариченко Б.Е. Теоретические основы информатики. – М.: Изд-во Горячая линия-Телеком, 2003. – 312 с.

файла на электронном носителе или высылается по электронной почте. В ряде работ предусмотрена автоматическая проверка корректности вычислений, что обеспечивает студенту дополнительные удобства в процессе самостоятельного выполнения заданий.

Мы будем весьма признательны, если отзывы по материалам практикума и предложения по его совершенствованию Вы направите по адресу:

620017, РОССИЯ, г. Екатеринбург, пр. Космонавтов, 26.
Уральский государственный педагогический университет,
кафедра НИТО. Mail: bes@uspu.ru

Авторы

Лабораторная работа 1. Энтропия. Информация.

Учебная задача:

Научиться оценивать энтропию и информацию опытов со случайными исходами.

Рабочий файл: ТОИ_Лр_1.xls

Необходимые теоретические сведения

Энтропия опыта с равновероятными исходами

$$H = \log_2 n, \quad (1.1)$$

где n – количество равновероятных исходов.

Энтропия опыта с n не равновероятными исходами

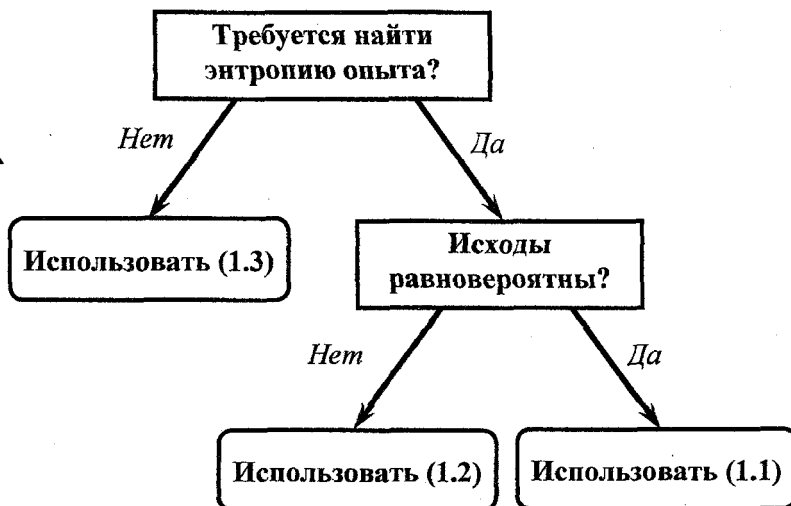
$$H = -\sum_{i=1}^n p_i \log_2 p_i, \quad (1.2)$$

где p_i – вероятность i -го исхода.

Энтропия отдельного исхода

$$H_i = -\log_2 p_i, \quad (1.3)$$

Общий алгоритм анализа хода решения задач:



Формулы (1.1)-(1.3), а также алгоритм могут быть использованы и при решении задач, связанных с нахождением количества информации.

Число вопросов с бинарными равновероятными ответами («да» – «нет») («выборочный каскад»), которые необходимо задать, для того чтобы полностью снять неопределенность опыта (ситуации):

$$k \geq \log_2 n \quad (1.4)$$

Поскольку количество вопросов выражается целым числом, при использовании формулы (4) k следует считать равным ближайшему целому числу, большему логарифма.

Перед началом работы рекомендуется создать на странице MS Excel следующую обобщенную схему расчетов:

Исход	N_i	p_i	H_i	$p_i H_i$
1				
2				
3				
...				
n				

$N_0 =$	1	$H =$	бит
---------	---	-------	-----

$=\text{СУММ}(C3:C7)$ $=\text{СУММ}(D3:D7)$ $=\text{СУММ}(F3:F7)$

$=D3 \cdot E3$
 $=-\text{LOG}(D3;2)$
 $=C3/\$C\8

В общем случае рассматривается опыт с n не равновероятными исходами. В приведенной схеме в колонке N_i указывается (вводится) количество состояний, связанных с i исходом. Суммирование N_i в ячейке **C8** позволяет определить общее число исходов N_0 . p_i – вероятность исхода i (обратите внимание, в ячейке **D8** найдена сумма вероятностей всех ис-

ходов; в случае правильных вычислений она равна 1). В колонке H_i – рассчитывается энтропия каждого из исходов по формуле (1.3). В колонке $p_i H_i$ рассчитываются «весовые» энтропии исходов. В ячейке **F8** по формуле (1.2) находится энтропия опыта.

Данная схема может быть применена для решения многих задач, связанных с вычислениями энтропии и количества информации.

Примеры решения задач

Задача 1. а) Чему равна энтропия, связанная с выпадением цифры «6» при броске кубика? б) Чему равна энтропия броска, если выпала цифра «6»?

Решение:

а) Поскольку речь идет об энтропии отдельного исхода (а не опыта в целом), согласно приведенного выше алгоритма выбираем для решения формулу (3). Для того чтобы ею воспользоваться, необходимо найти вероятность исхода; поскольку исходы равновероятны,

$$p = \frac{1}{n} = \frac{1}{6}$$

Следовательно, энтропия исхода

$$H_6 = -\log_2 p = \log_2 6 = 2,585 \text{ бит}$$

б) Используем описанную схему решения в MS Excel.

Исход	N_i	p_i	H_i	$p_i H_i$
«1»-«5»	5	0,833	0,263	0,219
«6»	1	0,167	2,585	0,431
$N_0=$	6	1,000	$H_0=$	0,650

Таким образом, энтропия броска (опыта) оказывается равной $H_0 = 0,650$ бит.

Задача 2. В корзине находится 25 шаров: 3 – красных, 8 – белых, 9 – желтых, 5 – синих. Опыт состоит в извлечении случайным образом одного шара. Какое количество информации связано с исходом опыта?

Решение:

Поскольку речь идет об информации, связанной с опытом, исходы которого не равновероятны, воспользуемся расчетной схемой в Excel.

Исход	N_i	p_i	I_i	$p_i I_i$
красный	3	0,12	3,059	0,367
белый	8	0,32	1,644	0,526
желтый	9	0,36	1,474	0,531
синий	5	0,20	2,322	0,464
$N_0=$	25	1,00	$I_0=$	1,888

Задача 3. Сколько вопросов с ответами «Да-Нет» необходимо задать человеку, чтобы узнать месяц его рождения?

Решение:

Количество возможных исходов $n = 12$; при условии, что рождение человека может произойти в любом месяце, их следует считать равновероятными – это позволяет воспользоваться формулой (1.4). Следовательно, количество вопросов

$$k \geq \log_2 n \geq \log_2 12 = 4$$

Порядок выполнения работы

- 1) Скопируйте в свою рабочую папку файл **ТОИ_Лр_1.xls** и откройте его. Если требуется, установите **средний** уровень безопасности макросов (меню **Сервис** → **Макрос** → **Безопасность**). Зарегистрируйтесь на листе **<Энтропия>**. На листе **<Общая схема>** реализуйте приведенную выше расчетную схему для 4-6 элементов (исходов) (например, для рассмотренной выше Задачи 2).
- 2) Решение задач, в которых требуется определить энтропию или количество информации следует начинать с анализа условия по представленному выше алгоритму.
- 3) Решите задачи, приведенные на страницах **<Энтропия>** и **<Информация>**. При необходимости таблицу со схемой решения копируйте с листа **<Общая схема>**.
- 4) Задания, в которых требуется произвести доказательства, выполняются в рабочих тетрадях (конспектах); они могут быть помещены в отчет с помощью редактора формул (по желанию студента).
- 5) При выполнении задания Э5 исходная функция должна быть записана аналитически, представлена в табличном виде (протабулирована) с 20-25 промежуточными значениями, по которым и должен быть построен график.

- 6) Отчет сохраните в рабочей папке. По завершении работы предъявите его преподавателю.

Контрольные вопросы:

1. Почему в определении энтропии как меры неопределенности выбрана логарифмическая зависимость между H и n ? Почему выбран логарифм по основанию 2?
2. Обоснуйте предложенную в работе схему решения задач.
3. Почему в формуле (1.4) используется знак « \geq »? При каких условиях его можно заменить знаком « $=$ »?
4. Объясните, почему максимум энтропии опыта в задаче Э5 приходится на $p = 0,5$?
5. В задаче И2 получилось, что с отгадыванием двузначного числа по цифрам и того же числа в целом связано одинаковое количество информации. Однако, число вопросов с ответами «да-нет» в этих двух методах отгадывания нужно задать разное. Почему?
6. Прокомментируйте результат решения задачи И5.
7. Прокомментируйте результат решения задачи И10.

Лабораторная работа 2.

Исследование статистических характеристик текста

Учебная задача:

Определить некоторые статистические характеристики заданного текста.

Рабочие файлы: ТОИ_Лр_2.xls, Generator.doc

Необходимые теоретические сведения

Сообщения, в которых вероятность появления любого отдельного знака алфавита в любом месте сообщения остается неизменной и не зависит от того, какие знаки предшествовали данному, называется **шенноновскими**.

Сообщения, в которых существует статистические связи между знаками или их сочетаниями, называются **сообщениями с памятью** или **марковскими сообщениями**.

Для того чтобы проверить, является ли некоторое сообщение **марковским** или **шенноновским**, необходимо произвести частотное исследование текста и определить вероятности (относительные частоты) каждого из N знаков алфавита $p_i (i=1 \dots N)$, а также вероятности всех возможных сочетаний двух букв $p_{ik}^{(fact)} (i, k = 1 \dots N)$ (при учете только двухбуквенных корреляций). Если для всех пар ik случайное событие – появление в тексте знака a_i – не оказывает влияния на другое случайное событие появление после него знака a_k (т.е. эти события являются независимыми) то должно выполняться условие:

$$p_{ik}^{(fact)} = p_i p_k \quad (2.1)$$

– в этом случае может быть сделано заключение, что сообщение является шенноновским. При практических оценках следует учитывать, что из-за статистической погрешности равенство в формуле (2.1) может оказаться нестрогим.

Если же условия (2.1) не выполняются хотя бы для одной пары ik , данное сообщение должно быть признано марковским.

Таким образом, анализ текста требует определения вероятностей всех входящих в него букв, а также определения вероятностей всех возможных парных сочетаний букв.

Порядок выполнения работы:

- 1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр_2.xls** и **Generator.doc**, откройте их. Если требуется, установите **средний** уровень безопасности макросов (меню Сервис→Макрос→Безопасность) в MS Word и MS Excel. Зарегистрируйтесь на листе <Статистика>.
- 2) В документе **Generator** в правое окно введите любые 3 символа (алфавит), в левое – объем текста (8000-15000); нажмите экранную клавишу [**Генерация**]. Сохраните сгенерированный текст в рабочую папку, добавив в имя файла свою фамилию.
- 3) Используя функции поиска и замены текстового редактора MS Word, найдите количество вхождения каждого из трех символов в тексте. Результаты занесите в табл. 1 отчета.
- 4) Дополните табл. 1, определите среднюю информацию на знак исходного алфавита.
- 5) В *Задании 2* введите все возможные двухбуквенные сочетания в табл. 2. Производя анализ текста, определите вероятности появления всех двухбуквенных сочетаний, результаты занесите в табл. 2 (*следует принять, что соседом последнего знака является первый знак, т.е. общее число двухбуквенных сочетаний должно совпадать с количеством знаков в исходном тексте*).
- 6) По данным табл. 1 рассчитайте $(P_{ij})_{нез}$, найдите разности с P_{ij} . Сделайте заключение о характере текста (шенноновский, марковский).
- 7) Считая двухбуквенные сочетания знаками нового алфавита, дополните табл. 2 и определите среднюю информацию на знак этого алфавита. Сопоставьте с данными из *Задания 1*, сделайте выводы.
- 8) Отчет сохраните в рабочей папке. По завершении работы предъявите его преподавателю.

Контрольные вопросы:

1. Предложите несколько способов подсчета количества сочетаний знаков в исходном тексте.
2. Почему рекомендуется генерировать текст длиной несколько тысяч знаков? Имеет ли смысл увеличивать эту длину, например, до сотен тысяч?
3. До какого знака после запятой должны совпадать значения P_{ij} и $(P_{ij})_{\text{нез}}$ в шенноновском сообщении? Почему? Чем определяется такая точность?
4. Почему при совпадении значений P_{ij} и $(P_{ij})_{\text{нез}}$ сообщение считается шенноновским?
5. Что изменится в выполнении работы, если алфавит текста будет содержать иное количество знаков (например, 4 или это будет алфавит естественного языка (например, русского))?
6. Что изменится в расчетах, если потребуется учесть трехбуквенные сочетания?
7. Прокомментируйте соотношение средних информации на знак, полученных в заданиях 1 и 2.
8. Зависит ли вывод, полученный в п. 7, от характера сообщения (шенноновское, марковское)? Почему?

Лабораторная работа 3.

Сравнение методов кодирования информации

Учебная задача:

Освоить различные методы кодирования текстовой информации, сравнить их эффективность.

Краткие теоретические сведения

Пусть имеется первичный алфавит A , содержащий N знаков с вероятностями их появления в тексте p_i .

Избыточность первичного двоичного кода определяется по формуле:

$$Q(A, 2) = \frac{K(A, 2)}{I_1^{(A)}} - 1, \quad (3.1)$$

где $K(A, 2)$ – средняя длина кодовой комбинации;

$I_1^{(A)}$ – средняя информация на знак первичного алфавита.

Для нахождения $K(A, 2)$ необходимо знать вероятности появления знаков первичного алфавита p_i , а также длины кодовых комбинаций каждого знака k_i ; тогда:

$$K(A, 2) = \sum_{i=1}^N k_i p_i \quad (3.2)$$

Средняя информация на знак первичного алфавита может быть найдена по формуле Шеннона:

$$I_1^{(A)} = - \sum_{i=1}^N p_i \log_2 p_i \quad (3.3)$$

Таким образом, выполнение работы предполагает нахождение по (3.3) для заданного алфавита $I_1^{(A)}$ (очевидно, эта величина будет одинаковой для всех методов кодирования данного алфавита), осуществление кодирования и определение k_i , вычисление по (3.2) средней длины кода и затем определение избыточности по формуле (3.1).

С методами двоичного кодирования можно познакомиться в пособии [1] или электронном курсе [3].

Часть 1. Методы алфавитного кодирования

Рабочий файл: **ТОИ_Лр_3(1).xls**

Порядок выполнения работы:

- 1) Скопируйте в свою рабочую папку файл **ТОИ_Лр_3(1).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе **<Отчет>**. Сохраните файл в рабочей папке со своей фамилией.

Важно: перед применением методов кодирования в заданиях 2-4 алфавиты должны быть упорядочены в порядке убывания вероятности знаков.

- 2) Для заданного алфавита построить неравномерный код с разделителем. Определите избыточность кода.
- 3) Для заданного алфавита построить неравномерный код Шеннона-Фано. Определите избыточность кода.
- 4) Для заданного алфавита построить неравномерный код Хаффмана. Определите избыточность кода.
- 5) Для заданного алфавита построить равномерный код. Определите избыточность кода.
- 6) Сопоставьте эффективности полученных кодов и сделайте заключение относительно использованных методов кодирования.
- 7) Отчет сохраните в своей рабочей папке. По завершении работы предъявите его преподавателю.

Контрольные вопросы:

1. Зачем необходимы разделители кодов при неравномерном кодировании? В каких случаях разделители не требуются?
2. Что такое префиксные коды? В чем смысл условия Фано?
3. Чем определяется длина равномерного кода для заданного алфавита? Как она соотносится со средней длиной кодов при кодировании по методу Хаффмана?
4. Всегда ли эффективность равномерного кодирования оказывается ниже по сравнению с эффективностью метода Хаффмана?

Часть 2. Сопоставление алфавитного и блочного кодирования

Рабочие файлы: ТОИ_Лр_3(2).xls, отчет по Лр 2.

Порядок выполнения работы:

- 1) Скопируйте в свою рабочую папку файл **ТОИ_Лр_3(2).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе **<Отчет>**. Нажмите экранную кнопку **<Загрузить отчет ЛР2>** – найдите и укажите свой файл с отчетом по Лабораторной работе № 2 – он будет перенесен в новый отчет; сохраните файл в рабочей папке со своей фамилией.
- 2) Скопируйте с листа **<ЛР_2>** данные о знаках и их вероятностях первичного алфавита и двухбуквенного алфавита, перенесите их в соответствующие заготовки таблиц на листе **<Отчет>**.
- 3) Произведите кодирование по методу Хаффмана первичного (из 3-х букв) алфавита. Определите избыточность кода.
- 4) Произведите кодирование по методу Хаффмана блоков, составленных из 2-х-буквенных сочетаний знаков исходного алфавита в **Задании 2**. Определите избыточность кода.
- 5) Сопоставьте полученные результаты, сделайте выводы.
- 6) Отчет сохраните в своей рабочей папке. По завершении работы предъявите его преподавателю.

Контрольные вопросы:

1. В чем значение первой теоремы Шеннона для кодирования?
2. Чем определяется минимальная средняя длина алфавитного кода? При каком методе кодирования она достигается?
3. Существует ли предел снижения избыточности кода путем построения блочных кодов? Если «да», то чем он определяется?
4. Почему избыточность блочного кода оказывается меньше избыточности алфавитного кода, если в обоих случаях использовался метод Хаффмана?
5. Как соотносятся средние информации на знак при алфавитном и блочном кодировании? Почему?

Лабораторная работа 4.

Кодирование числовой информации

Учебная задача:

Освоить методы представления и кодирования числовой информации.

Часть 1. Системы счисления

Рабочий файл: ТОИ_Пр_4.xls.

Краткие теоретические сведения

Система счисления – это правило записи чисел с помощью заданного набора специальных знаков – цифр.

Любое целое число Z может быть представлено в позиционной системе счисления с основанием p в виде многочлена:

$$Z_p = a_{k-1} \cdot p^{k-1} + a_{k-2} \cdot p^{k-2} + \dots + a_1 \cdot p^1 + a_0 \cdot p^0 = \sum_{j=0}^{k-1} a_j \cdot p^j, \quad (4.1)$$

где коэффициенты разложения a_j удовлетворяют условию:

$$0 \leq a_j \leq p - 1$$

Аналогично, любая правильная дробь $0, Y$ также может быть представлена в виде многочлена с отрицательными показателями степеней:

$$0, Y_p = a_1 \cdot p^{-1} + a_2 \cdot p^{-2} + \dots + a_m \cdot p^{-m} + \dots = \sum_{j=1}^k a_j \cdot p^{-j} \quad (4.2)$$

Одно и то же число может быть представлено в различных системах счисления. В связи с этим возникает вопрос о порядке перевода числа, представленного в одной системе счисления (СС), в СС с другим основанием.

Перевод целых десятичных чисел Z в СС с основанием p может быть реализован посредством рекуррентных соотношений:

$$\begin{aligned} Z_0 &= Z_{(10)}; \\ Z_{i+1} &= Z_i \operatorname{div} p; \quad a_i = Z_i \bmod p, \end{aligned} \quad (4.3)$$

где посредством функции div находится результат целочисленного деления, а функция \bmod дает остаток от целочисленного деления. В пакете MS Excel эти результаты могут быть получены посредством функций ЦЕЛОЕ(Z_i/p) и ОСТАТ(Z_i/p).

Процедура прекращается на том шаге, когда оба значения – Z_{i+1} и a_i – окажутся равными 0. Полученное число считывается в обратном порядке.

Перевод простых десятичных дробей $0, Y$ в СС с основанием p может быть реализован посредством следующих рекуррентных соотношений:

$$\begin{aligned} Y_0 &= 0, Y_{(10)}; \\ a_i &= \text{int}(Y_i \cdot p); Y_{i+1} = Y_i \cdot p - a_i, \end{aligned} \quad (4.4)$$

где функция int означает нахождение целой части числа путем отбрасывания дробной (без округления). В MS Excel результат может быть получен использованием $\text{ЦЕЛОЕ}(Y_i \cdot p)$.

Следует сознавать, что при переводе дробей может возникнуть ситуация, когда конечная дробь в 10-ной СС превратиться в бесконечную дробь в СС с основанием p или наоборот. Поэтому процедура прекращается на том шаге, когда оба значения – Y_i и a_i – окажутся равными 0 либо по достижении желательной точности вычислений.

Перевод целых чисел Z_p в 10-ю СС может быть реализован на основании представления исходного числа в форме (4.1) и последующего проведения вычислений по правилам 10-ной арифметики.

Аналогично, *перевод простых дробей $0, Y_p$ в 10-ю СС* может быть реализован на основании представления исходного числа в форме (4.2) и последующего проведения вычислений по правилам 10-ной арифметики.

Порядок выполнения работы:

- 1) Скопируйте в свою рабочую папку файл **ТОИ_Лр_4.xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе **<Отчет_1>**; сохраните файл в рабочей папке со своей фамилией.
- 2) Ознакомьтесь с реализацией алгоритма перевода (4.3) на листе **<Перевод СС>**. Выполните упражнения (1.а) в *Задании 1*. При необходимости модернизируйте расчетную схему.

- 3) Реализуйте в MS Excel алгоритм (4.1) перевода $Z_p \rightarrow Z_{10}$. Выполните упражнения (1.b) в *Задании 1*.
- 4) Ознакомьтесь с реализацией алгоритма перевода (4.4) на листе <Перевод СС>. Выполните упражнения (2.a) в *Задании 2*. При необходимости модернизируйте расчетную схему.
- 5) Реализуйте в MS Excel алгоритм (4.2) перевода $\theta, Y_p \rightarrow \theta, Y_{10}$. Выполните упражнения (2.b) в *Задании 2*.
- 6) С помощью построенных расчетных схем выполните упражнения по переводу смешанных чисел в *Задании 3*.
- 7) Выполните преобразования в *Задании 4*.
- 8) Проверьте полученные результаты, нажав экранную кнопку [Проверка]. При наличии ошибок исправьте их. Сохраните отчет в своей рабочей папке. По завершении работы предъявите его преподавателю.

Контрольные вопросы:

1. В чем отличие аддитивных и аддитивно-мультипликативных СС? Приведите примеры.
2. Можно ли унарную систему счисления считать позиционной СС с основанием 1?
3. Почему при переводе дробного числа из одной системы счисления в другую конечная дробь может стать бесконечной или наоборот? Приведите примеры.
4. Каким образом перевести число в естественной форме представления (имеются целая и дробная части) из одной десятичной СС в другую десятичную?
5. Сформулируйте правило перевода целых и дробных чисел между СС, основание одной из которых является целочисленной степенью другого основания.
6. Какая СС является наиболее экономичной при представлении чисел? Как это доказать?
7. Усовершенствуйте реализацию алгоритмов, представленных на листе <Перевод СС>, таким образом, чтобы в некоторой ячейке сразу выдавался результат перевода (как единое число).

Часть 2. Нормализация чисел

Рабочий файл: ТОИ_Пр_4.xls.

Краткие теоретические сведения

Число X_p называется нормализованным в системе счисления p , если оно представлено в виде:

$$X_p = \pm M_p \cdot p^{\pm k_p}, \quad (4.5)$$

где M_p – мантисса нормализованного числа, удовлетворяющая условию:

$$\frac{1}{p} \leq M_p < 1, \quad (4.6)$$

а k_p – порядок нормализованного числа, представленный в СС p .

Например,

$$(3213202,12)_4 = 0,321320212_4 \cdot 4^{13_4}$$

$$(0,0000004321)_5 = 0,4321_5 \cdot 5^{-11_5}$$

Для перевода десятичного числа в естественной форме представления в нормализованное число в СС p можно сначала перевести число в естественной форме из 10 в p , пользуясь описанными выше алгоритмами, а затем нормализовать его в СС p .

Например, необходимо представить число $16,5_{10}$ в нормализованной форме в 8-ричной СС. Пользуясь алгоритмами (4.3) и (4.4), получаем: $17,375_{10} = 21,3_8 = 0,213_8 \cdot 8^2$; таким образом, $M_8 = 0,213$; $k_8 = 2$.

Для перевода числа из естественной формы представления в СС p в нормализованное число в СС q можно сначала перевести число в естественной форме из p в 10-ю СС по алгоритмам (4.1) и (4.2), затем в естественную форму в СС q , а затем нормализовать его в СС q .

Преобразование числа из нормализованной формы в СС p в нормализованное число в СС q удобнее осуществлять через промежуточный переход к 10-ой СС.

Порядок выполнения работы:

- 1) Откройте из своей рабочей папки файл с отчетом по лабораторной работе **ТОИ_Лр_4.xls**. Если требуется, установите **средний** уровень безопасности макросов. Перейдите на лист **<Отчет_2>**.
- 2) Используя реализованные в предыдущей части данной ЛР алгоритмы перевода целых и дробных чисел в различные СС, выполните *Задания 1-2(а-с')*. Результаты занесите в соответствующие таблицы.
- 3) Выполните *задание 3а'*.
- 4) Исследуйте порядок преобразования нормализованных чисел между СС, основание одной из которых является целочисленной степенью основания другой СС в задании 3b (на примере преобразований $2 \leftrightarrow 8 \leftrightarrow 16$). Преобразование должно осуществляться без перехода к 10-ой СС в качестве промежуточной. Сформулируйте обобщенное правило подобных преобразований.
- 5) Выполните проверку правильности выполнения заданий. Отчет сохраните и предъявите преподавателю.

Контрольные вопросы:

1. В чем преимущества и недостатки представления чисел в нормализованной форме?
2. Почему мантисса нормализованного числа должна удовлетворять условию (4.6)?
3. Сформулируйте правило построения нормализованного числа из числа в естественной форме представления для произвольной СС.
4. Сформулируйте правило построения нормализованного десятичного числа из числа в естественной форме представления в произвольной СС.
5. Сформулируйте обобщенное правило преобразований нормализованных чисел между СС, основание одной из которых является целочисленной степенью другого основания.

1 Рекомендация: при выполнении заданий 2с и 3а преобразования необходимо осуществлять через десятичную СС в качестве промежуточной.

Часть 3. Операции с кодами чисел

Рабочий файл: ТОИ_Пр_4.xls.

Краткие теоретические сведения

Для представления двоичных чисел в памяти и регистрах компьютера отводится конечное число разрядов. По этой причине можно говорить, что в компьютере производится обработка **кодов чисел**, которые представляют собой запись чисел в ограниченной разрядной сетке.

При выполнении данной лабораторной работы будет использоваться 16-ти-разрядная сетка. Нумерация разрядов сетки осуществляется справа налево, младший разряд имеет номер «0» (соответственно, старший разряд – номер «15»).

При записи **целых чисел без знака** (положительных) заполнение сетки производится, начиная с младшего разряда. Очевидно, максимальное число, которое может быть записано в 16-ти-разрядную сетку, будет равно $2^{16} - 1 = 65535$ (во всех ячейках стоят 1). При попытке записать большее по величине число возникает ошибочная ситуация, называемая *переполнением* (число выходит за установленную сетку).

Запись **целых чисел со знаком** в отведенную разрядную сетку осуществляется в *дополнительном коде*. Для двоичных целых чисел он строится по следующим правилам:

- для $Z_2 \geq 0$ дополнительный код совпадает с самим числом ($DK = Z_2$);
- для $Z_2 < 0$ дополнительный код совпадает с дополнением модуля числа, т.е. $DK = D(|Z_2|, k)$.

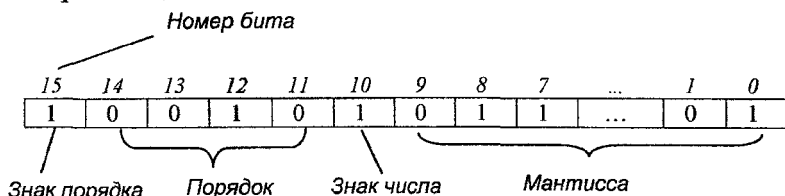
В свою очередь, дополнение модуля двоичного числа строится следующим образом:

- инвертировать представление исходного числа в отведенной разрядной сетке;
- к последнему (младшему) разряду инвертированного представления прибавить 1 по правилам двоичной арифметики.

Таким образом, 15-й (старший) бит целого положительного числа всегда содержит 0, а отрицательного числа 1.

Очевидно, в такой форме представления максимальным по модулю будет число $2^{15} - 1 = 32767$ – при его превышении возникает переполнение.

При размещении **двоичного нормализованного числа** запись всех составляющих (знак числа, мантисса, знак порядка и порядок) осуществляется в строго определенные ячейки разрядной сетки. Поскольку в данной лабораторной работе используется 16-ти-разрядная сетка, примем следующий порядок размещений в ней вещественных чисел:



Поскольку значение мантиссы лежит в интервале $0,1_2 \leq M_2 < 1$, ноль в разряде целых и разделитель десятичных разрядов в представлении не включается, т.е. мантисса содержит только цифры дробной части.

Таким образом, максимальный порядок числа в данной форме представления равен $2^4 - 1 = 15$; следовательно, максимальным оказывается число $2^{15} = 32767_{10}$.

Поскольку соотношение между числом разрядов в двоичном и десятичном представлении числа: $k_2 = 3,322 \cdot k_{10}$, 10-битная двоичная мантисса обеспечивает точность представления и вычисления до 3-х знаков в десятичной мантиссе – все разряды, следующие за тысячными, будут пропадать, что приводит к погрешности записи и обработки вещественных чисел.

Операции с двоичными числами выполняются согласно правилам двоичной арифметики:

Сложение производится согласно таблице сложения, которая для двоичных чисел имеет вид:

$$\begin{array}{ll}
 0 + 0 = 0 & 0 + 1 = 1 \\
 1 + 0 = 1 & 1 + 1 = 10
 \end{array}$$

В последнем случае в том разряде, где находились слагаемые, оказывается 0, а 1 переносится в старший разряд.

Умножение производится согласно таблице умножения:

$$\begin{array}{ll} 0 \cdot 0 = 0 & 0 \cdot 1 = 0 \\ 1 \cdot 0 = 0 & 1 \cdot 1 = 1 \end{array}$$

Операции **сложения и умножения** являются двуместными, поскольку выполняются над двумя числами и, следовательно, требуют два регистра для записи обрабатываемых чисел и один для записи результата.

Над целыми числами со знаком определены операции сложения и умножения. Операция вычитания как самостоятельная отсутствует, поскольку она эквивалентна сложению с дополнительным кодом вычитаемого. Операция деления над целыми числами не определена.

Сложение и вычитание нормализованных чисел возможно только при совпадении их порядков. Поэтому перед выполнением операции должен быть произведен сдвиг мантиссы меньшего слагаемого в разрядной сетке таким образом, чтобы порядки совпали. Сложение разрядов мантиссы осуществляется побитно; при необходимости результат сложения нормализуется.

Умножение нормализованных чисел производится следующим порядком:

- сложить порядки сомножителей;
- перемножить мантиссы;
- при необходимости нормализовать результат умножения.

Деление нормализованных чисел эквивалентно умножению на число, у которого изменен на противоположный знак порядка.

Порядок выполнения работы:

- 1) Откройте из своей рабочей папки файл с отчетом по лабораторной работе **ТОИ_Лр_4.xls**. Если требуется, установите **средний** уровень безопасности макросов. Перейдите на лист **<Отчет_3>**.
- 2) Используя реализованные ранее алгоритмы перевода целых и дробных чисел в различные СС, выполните *Задание 1(a-b)*. Результаты занесите в соответствующие таблицы. При возникновении переполнения сделайте отметку («1») в соответствующей ячейке таблицы.

Рекомендация: перед записью десятичных чисел в разрядную сетку удобнее перевести их в 8-ричную СС.

- 3) При выполнении *Задания 1с* следует разместить слагаемые в двух первых строках таблицы, а результат – в третьей.

Важно: 1) записью чисел в разрядную сетку должна производиться в дополнительном коде;

2) Появление «1» в Дополнительном разряде в операции вычитания (сложения с дополнительным кодом) не является переполнением и не свидетельствует об ошибке выполнения операции.

- 4) При выполнении *Задания 2* нужно использовать установленное выше распределение разрядов для записи нормализованного двоичного числа. Для удобства десятичные числа рекомендуется сначала перевести в 8-ричную СС.
- 5) Ознакомьтесь с примером выполнения *Задания 2с*. В нем, в частности, показано, что умножение вещественных нормализованных двоичных чисел сводится к операциям сдвига и сложения. По аналогии проведите предложенные вычисления.
- 6) Приведите примеры, показывающие, что при использовании конечной разрядной сетки для представления чисел может возникнуть погрешность вычислений.
- 7) Сохраните отчет. Предъявите его преподавателю.

Контрольные вопросы:

1. Что такое «код числа»? Какие коды применяются в компьютерах?
2. Чем различается представление целых чисел со знаком и без знака в конечной разрядной сетке?
3. Сформулируйте правило построения дополнительного кода числа.
4. Как представляется вещественное число в конечной разрядной сетке? Чем определяется точность представления и обработки чисел?
5. В чем преимущества и недостатки кодирования вещественных чисел с помощью нормализованной двоичной формы?
6. Опишите порядок действий при сложении (вычитании) двух нормализованных двоичных чисел в конечной разрядной сетке.
7. Опишите порядок действий при умножении (делении) двух нормализованных двоичных чисел в конечной разрядной сетке.
8. В каких случаях при обработке кодов вещественных чисел возникает погрешность вычислений? Почему? Может ли погрешность не возникать?

Лабораторная работа 5.

Исследование дискретных двоичных каналов передачи информации

Учебная задача:

На основе экспериментирования с компьютерными моделями двоичных каналов выявить их характеристики.

Краткие теоретические сведения

Дискретный канал – канал связи, используемый для передачи дискретных сообщений.

Дискретный канал называется **двоичным**, если он используется для передачи знаков двоичного алфавита.

Дискретный канал называется **однородным**, если характеризующие его апостериорные вероятности не изменяются с течением времени.

Дискретный двоичный канал называется **симметричным**, если для апостериорных вероятностей выполняются условия:

$$p_0(1) = p_1(0) = p; \quad p_0(0) = p_1(1) = 1 - p,$$

где p – вероятность возникновения ошибки при передаче знака («0» или «1») по каналу.

Пропускная способность дискретного канала без помех численно равна максимальной (или несущей) частоте полосы пропускания:

$$C = \nu_m \quad (5.1)$$

Пропускная способность однородного двоичного симметричного канала с помехами может быть вычислена:

$$C = \nu \cdot [1 + (1 - p) \log_2(1 - p) + p \log_2 p], \quad (5.2)$$

где p – вероятность ошибки передачи, ν – частота передачи.

Дискретный двоичный канал **называется каналом со стиранием**, если на приемном конце канала может появиться знак, который не может быть интерпретирован ни как «0», ни как «1» (это эквивалентно стиранию исходного знака и появлению нового, которого не было в начальном двоичном алфавите на входном конце канала).

Пропускная способность однородного двоичного симметричного канала со стиранием может быть вычислена:

$$C = L \cdot \{(1-q)[1 - \log_2(1-q)] + (1-p-q) \cdot \log_2(1-p-q) + p \cdot \log_2 p\}, \quad (5.3)$$

где p – вероятность возникновения ошибки при передаче знака (замена «0» на «1» или «1» на «0»); q – вероятность стирания (замена «0» или «1» знаком стирания); ν – частота передачи.

Часть 1. Двоичный канал без стирания

Рабочий файл: *ТОИ_Лр_5.xls*

Порядок выполнения работы:

- 1) Скопируйте в свою рабочую папку файл **ТОИ_Лр_5.xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе <**Двоичный канал**>. Сохраните файл в рабочей папке со своей фамилией.
- 2) При выполнении *Задания 1* предполагается экспериментирование с экранной моделью двоичного канала связи. По каналу по очереди должны передаваться элементарные сигналы («0» и «1»). Количество сигналов задается согласно таблицам в окне «**Число сигналов**»; после этого следует нажать экранную кнопку [**Передача «0»**] или [**Передача «1»**]. Результаты заносятся и обрабатываются в таблицах. Очистка окон производится нажатием кнопки [**Очистка**].
- 3) После занесения в таблицы результатов измерений необходимо найти средние вероятности искажения сигналов при передаче – отдельно $\langle p_0 \rangle$ и $\langle p_1 \rangle$. Далее нужно сопоставить эти величины – если их различие находится в пределах нескольких тысячных, то делается вывод о симметричности канала. В этом случае за вероятность искажения сигнала p принимается средняя величина $\langle p_0 \rangle$ и $\langle p_1 \rangle$ – именно она используется далее при расчете пропускной способности канала в *Задании 2*.

- 4) По завершении расчетов в *Задании 2* нужно нажать экранную клавишу [**Проверка**] – если выявятся ошибки, вычисления (или измерения!) следует произвести заново.
- 5) В *Задании 3* следует построить граф канала связи (используя встроенный графический редактор MS Excel).
- 6) При выполнении *Задания 4* необходимо протабулировать функцию

$$\frac{C}{v}(p)$$

для симметричного двоичного канала в интервале $p \in [0,1]$ (20-25 точек) и построить ее график.

- 7) Сформулируйте и запишите выводы по работе. Сохраните отчет. Представьте его преподавателю.

Контрольные вопросы:

1. Тожественны ли понятия «канал связи» и «линия связи»? Ответ обоснуйте.
2. Верно ли, что характер канала (дискретный или аналоговый) определяется свойствами среды, в которой происходит передача сигналов?
3. Перечислите основные характеристики дискретного канала связи.
4. Каким образом должны соотноситься скорость передачи информации и пропускная способность канала связи, чтобы при передаче не происходило потери информации?
5. В чем состоит влияние помех на процесс передачи информации по дискретному каналу связи? Влияют ли помехи на пропускную способность канала?
6. Каким образом в выполненной лабораторной работе доказывается, что канал является симметричным? Зачем это делается?
7. Что такое «граф канала»? Что он отражает? Каковы правила его построения?
8. Почему при вероятности искажения сигнала $p = 1$ пропускная способность симметричного канала оказывается такой же, как у канала без помех?

Часть 2. Двоичный канал со стиранием

Рабочий файл: ТОИ_Лр_5.xls

Порядок выполнения работы:

- 1) Откройте из своей рабочей папки файл с отчетом по 1-й части ЛР **ТОИ_Лр_5.xls**. Если требуется, установите **средний** уровень безопасности макросов. Перейдите на лист <Канал со стиранием>.
- 2) Проведите исследования канала с целью определения вероятностей искажения (p) и стирания (q). Знак стирания обозначен «s». Убедитесь, что канал является симметричным.
- 3) Проведите вычисления пропускной способности канала. Проверьте правильность расчетов. Проведите сопоставление полученного значения с результатами из Части 1 данной ЛР, а также с пропускной способностью канала без помех.
- 4) Постройте граф канала.
- 5) Для полученного значения q постройте график зависимости
$$\frac{C}{v}(p, q = \text{Const})$$

Сопоставьте полученный характер зависимости с аналогичной зависимостью из Части 1 данной ЛР.

- 6) Сформулируйте и запишите выводы по работе. Сохраните отчет. Представьте его преподавателю.

Контрольные вопросы:

1. Что такое «стирание» при передаче сигналов? Почему оно происходит?
2. Как соотносятся вероятности искажения (p) и стирания (q) для дискретного канала связи? Почему?
3. Каким образом наличие стирания влияет на пропускную способность канала связи? Почему?
4. Как влияет стирание на характер зависимости пропускной способности от вероятности искажения сигнала? Предложите объяснение этому влиянию.

Лабораторная работа 6.

Построение помехоустойчивого кода

Учебная задача:

По заданной производящей матрице (проверочной матрице) построить помехоустойчивый систематический код. Определить характеристики кода.

Пример выполнения заданий

Пусть имеется производящая матрица систематического кода:

$$G_{7,4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

По данной исходной матрице необходимо выполнить следующие построения:

- 1) Определить характеристики кода (n , k , S , S_p).
- 2) Построить все разрешенные кодовые комбинации.
- 3) Определить минимальное кодовое расстояние и по нему установить возможности кода по обнаружению и исправлению ошибок.
- 4) Составить уравнения проверок.
- 5) Построить схему кодера.
- 6) Построить проверочную матрицу.
- 7) Составить таблицу исправлений.
- 8) Построить схему декодера.

Выполнение заданий

По информационной части заданной производящей матрицы видно, что она представлена диагональной подматрицей, следовательно, код является *каноническим* и для его построения применимы матричные операции.

- 1) Из анализа структуры матрицы $G_{7,4}$ видно, что она содержит 7 столбцов и 4 строки. Следовательно, длина кодовой

комбинации $n = 7$; число информационных бит $k = 4$; их разность дает число проверочных бит $r = n - k = 3$. Таким образом, требуется построить канонический систематический код (7,4).

Общее количество кодовых комбинаций $S = 2^n = 128$.

Число разрешенных кодовых комбинаций $S_p = 2^k = 16$

2) Среди разрешенных кодовых комбинаций всегда имеется тривиальная $U^{(0)} = \{0000000\}$. Еще 4 кода могут быть записаны непосредственно из строк матрицы $G_{7,4}$:
 $U^{(1)} = \{1000110\}$, $U^{(2)} = \{0100101\}$, $U^{(3)} = \{0010011\}$,
 $U^{(4)} = \{0001111\}$.

Остальные 11 кодов получаются построением линейных комбинаций базисных кодовых векторов.

$$\begin{aligned} U^{(5)} &= U^{(1)} \oplus U^{(2)} = \{1100011\}; \\ U^{(6)} &= U^{(1)} \oplus U^{(3)} = \{1010101\}; \\ U^{(7)} &= U^{(1)} \oplus U^{(4)} = \{1001001\}; \\ U^{(8)} &= U^{(2)} \oplus U^{(3)} = \{0110110\}; \\ U^{(9)} &= U^{(2)} \oplus U^{(4)} = \{0101010\}; \\ U^{(10)} &= U^{(3)} \oplus U^{(4)} = \{0011100\}; \\ U^{(11)} &= U^{(1)} \oplus U^{(2)} \oplus U^{(3)} = U^{(5)} \oplus U^{(3)} = \{1110000\}; \\ U^{(12)} &= U^{(1)} \oplus U^{(2)} \oplus U^{(4)} = U^{(5)} \oplus U^{(4)} = \{1101100\}; \\ U^{(13)} &= U^{(1)} \oplus U^{(3)} \oplus U^{(4)} = U^{(6)} \oplus U^{(4)} = \{1011010\}; \\ U^{(14)} &= U^{(2)} \oplus U^{(3)} \oplus U^{(4)} = U^{(8)} \oplus U^{(4)} = \{011101\}; \\ U^{(15)} &= U^{(1)} \oplus U^{(2)} \oplus U^{(3)} \oplus U^{(4)} = U^{(11)} \oplus U^{(4)} = \{1111111\}. \end{aligned}$$

3) По минимальному количеству единиц в построенных кодовых комбинациях находится минимальное кодовое расстояние: $d_{\min} = 3$. В свою очередь, по нему можно установить кратности обнаруживаемой и исправляемой ошибок из условий:

$$\theta_{об} = d_{\min} - 1 = 2; \quad \theta_u = \text{int}\left(\frac{d_{\min} - 1}{2}\right) = 1.^1$$

¹ Функцией **int** в формуле обозначена операция нахождения целой части числа путем отбрасывания дробной части без округления.

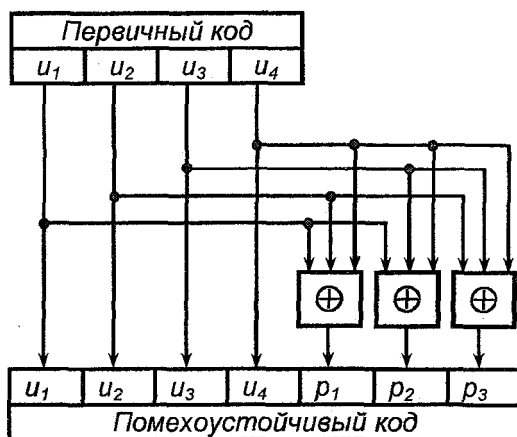
4) Для получения уравнений проверок нужно каждую строку (i) производящей матрицы домножить на u_i и произвести суммирование в столбцах:

$$\begin{pmatrix} u_1 & 0 & 0 & 0 & u_1 & u_1 & 0 \\ 0 & u_2 & 0 & 0 & u_2 & 0 & u_2 \\ 0 & 0 & u_3 & 0 & 0 & u_3 & u_3 \\ 0 & 0 & 0 & u_4 & u_4 & u_4 & u_4 \end{pmatrix}$$

Три последних столбца дают нужные уравнения проверок:

$$\begin{cases} p_1 = u_1 \oplus u_2 \oplus u_4 \\ p_2 = u_1 \oplus u_3 \oplus u_4 \\ p_3 = u_2 \oplus u_3 \oplus u_4 \end{cases}$$

5) Имея уравнения проверок можно построить схему кодирующего устройства (кодера).



6) Проверочную матрицу можно построить из уравнений проверок либо из производящей матрицы. Для этого нужно транспонировать проверочную подматрицу производящей матрицы:

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}^T = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Справа к ней пристраивается единичная диагональная размером (r, r) ; в рассматриваемом примере $(3, 3)$. Таким образом, полностью проверочная матрица будет выглядеть следующим образом:

$$H_{7,3} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

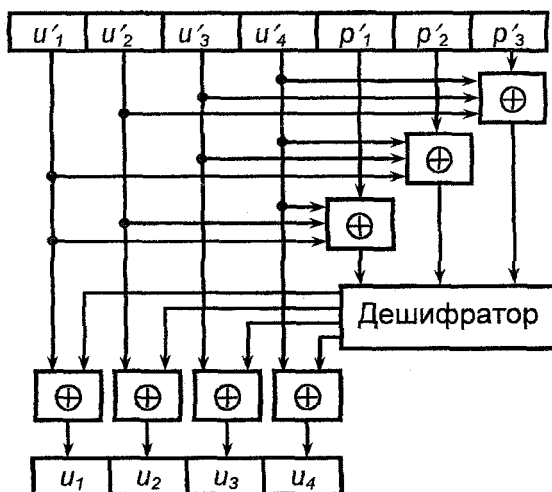
Ясно, что при необходимости из проверочной матрицы $H_{n,r}$ можно, произведя обратные действия, получить производящую $G_{n,k}$.

7) Таблица исправлений может быть построена из следующих соображений.

Предположим, что ошибочно передан бит u_i , а все остальные знаки кода переданы верно. Это скажется на значениях тех проверочных бит, в которые входит u_i — а нашем случае — p_1 и p_2 . Таким образом, искажению u_i будет соответствовать синдром ошибки $Q = (110)$. Здесь 1 — соответствует номеру проверочного уравнения, в котором фиксируется ошибка, а 0 — тому, где ошибки нет. Для исправления ошибки необходимо к ошибочному биту прибавить 1 по модулю 2, к остальным же битам не прибавлять ничего (или, что то же самое, прибавить 0). Исправлять ошибки в контрольных битах p_r , даже если они будут там обнаружены, при декодировании не требуется. Таким образом, получаем полную таблицу исправлений:

Синдром	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
Вектор ошибки	001	010	011	100	101	110	111
Ошибочная позиция	p_1	p_2	u_3	p_3	u_2	u_1	u_4
Выход дешифратора ошибок	0000	0000	0010	0000	0100	1000	0001

8) Непосредственно исправление ошибочного бита производит блок декодера, называемый *дешифратором*: если синдром указывает на ошибку в одном из проверочных бит, исправлений в информационных битах не требуется и дешифратор выдает 0 на все выходы; если синдром указывает на ошибку в каком-либо информационном бите, дешифратор посылает 1 на вход сумматора по модулю 2. Схема декодера систематического кода для построенного кода будет выглядеть следующим образом:



Задания выполнены.

Рабочий файл: ТОИ_Лр_6.xls

Порядок выполнения работы.

- 1) Скопируйте в свою рабочую папку файл **ТОИ_Лр_6.xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе **<Отчет>**. Сохраните файл в рабочей папке со своей фамилией.
- 2) После регистрации нажмите экранную кнопку **[Вариант]** — исходная матрица (G или H) появится в отчете.

- 3) Если исходной является производящая матрица G , выполните *Задания 1-8* в последовательности, описанной выше.
- 4) Если исходной является проверочная матрица H , задания выполняются в следующем порядке:
 - (1) Построить производящую матрицу.
 - (2) Определить характеристики кода (n, k, S, S_p) .
 - (3) Построить все разрешенные кодовые комбинации.
 - (4) Определить минимальное кодовое расстояние и по нему установить возможности кода по обнаружению и исправлению ошибок.
 - (5) Составить уравнения проверок.
 - (6) Построить схему кодера.
 - (7) Составить таблицу исправлений.
 - (8) Построить схему декодера.
- 5) Схемы кодера и декодера строятся с использованием встроенного графического редактора MS Excel
- 6) По завершении работы сохраните отчет; предъявите его преподавателю.

Контрольные вопросы:

1. Дайте определение помехоустойчивого кода. За счет чего код приобретает качество помехоустойчивости?
2. Какой помехоустойчивый код называется систематическим (линейным)?
3. Чем определяется избыточность помехоустойчивого кода? Можно ли ее уменьшить?
4. Что такое «*кодовое расстояние*»? Какое значение имеет эта величина для характеристик помехоустойчивости кода?
5. Почему матрица кода G называется производящей? Каковы правила ее построения?
6. Что такое «*синдром ошибки*»? Каким образом он устанавливается? Могут ли разные биты помехоустойчивого кода иметь одинаковые синдромы ошибок? Почему?
7. Каковы функции дешифратора в декодере?
8. Можно ли по схеме кодера (или декодера) установить все характеристики кода?

Лабораторная работа 7.

Изучение дискретных устройств обработки информации

Учебная задача:

Освоить методы построения и описания дискретных устройств обработки информации.

Рабочий файл: ТОИ_Лр_7.xls

Краткие теоретические сведения

Дискретные устройства обработки информации – абстрактные устройства, осуществляющие преобразование информации в дискретной форме представления по заданному алгоритму.

Дискретное устройство называется **двоичным**, если оно преобразует информацию, представленную двоичным алфавитом.

Примем следующую классификацию дискретных устройств обработки информации:

- устройства без памяти – *комбинационные схемы*;
- устройства с конечным числом ячеек памяти – *конечные автоматы*;
- устройства с бесконечной памятью (пример – машина Тьюринга).

Первые два типа устройств изучаются в данной лабораторной работе; исследование *машины Тьюринга* предусматривается в ЛР № 8.

Часть 1. Комбинационные схемы

Комбинационная схема может обеспечить обработку двоичных сигналов, подаваемых на ее вход, в соответствии с заданной логической функцией (системой функций), связывающей выходные сигналы с входными. Обработка производится комбинацией (схемой) *логических вентилях (логических элементов)*, реализующих три основные логические функции: И, ИЛИ и НЕ – они образуют *базис* элементов. При конструировании схем выходы вентилях могут присоединяться

ко входам других вентилях или быть вершинами (выходами) схемы; не может существовать соединений выхода данного вентиля с его входом (обратная цепь).

В общем случае комбинационная схема описывается совокупностью *трех* компонентов: алфавитом входных сигналов X , алфавитом выходных сигналов Y и *функцией* (системой функций) *выходов*, связывающих выходные сигналы с входными θ . Поскольку рассматриваются только двоичные устройства, входной и выходной алфавиты совпадают – это двоичный алфавит, содержащий знаки $\{0, 1\}$. Таким образом, двоичная комбинационная схема задается логической функцией (системой функций) *выходов* $\theta_j(x_1, x_2, \dots, x_N)$. Количество функций выхода равно числу *выходов* схемы ($y_j = \theta_j$).

Имеются три взаимосвязанные формы представления дискретных устройств без памяти: *графическая* (комбинационная схема), *функциональная* (система логических функций, реализуемый схемой), *табличная* (таблица значений на входах и выходах схемы – *таблица истинности*). По любой из указанных форм могут быть построены две другие.

В отношении комбинационных схем выделяется две группы задач: *анализа* и *синтеза*.

Задача анализа: по имеющейся схеме определить функции обработки (функции *выходов*) и построить таблицу значений.

Задачи синтеза: по заданным функциям обработки или таблице значений построить комбинационную схему.

Пример решения задачи **синтеза** комбинационной схемы.

Таблица истинности для некоторой схемы имеет вид:

x_1	x_2	y_1	y_2
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

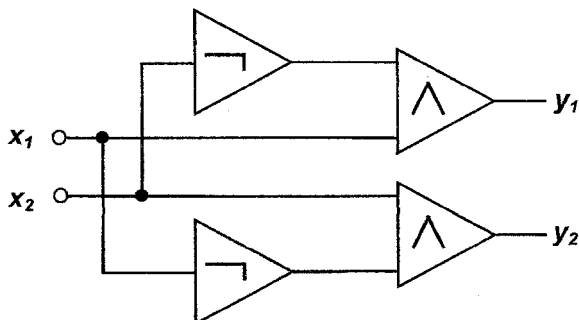
По данной таблице необходимо определить функции обработки (функции *выходов*), а также построить комбинационную схему.

Из представленной таблицы видно, что схема имеет два независимых входа (x_1 и x_2) и два выхода (вершины) – y_1 и y_2 . Анализируя исходную таблицу, можно усмотреть следующие соотношения, связывающие входные и выходные сигналы:

$$y_1 = \bar{x}_1 \wedge x_2;$$

$$y_2 = x_1 \wedge \bar{x}_2$$

Схема:



Задача решена.

Безусловно, наиболее сложным при решении задачи синтеза является определение вида функций обработки по имеющейся таблице истинности.

Порядок выполнения работы:

- 1) Откройте файл **ТОИ_Лр_7.xls**. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе **<Комбинационные схемы>**. Сохраните в рабочей папке со своей фамилией.
- 2) В *Задании 1а* нужно по схеме, мысленно задавая различные комбинации входных сигналов, определить сигнал на выходе и занести в таблицу. Необходимо записать аналитический вид функции выходов.
- 3) В *Задании 1б* требуется построить эквивалентную схему. Порядок выполнения: по исходной схеме построить функцию, упростить ее (пользуясь соотношениями математической логики – справочник можно вызвать экранной кнопкой), построить схему для упрощенной функции, составить таблицу значений, проверить, что для исходной схемы она также справедлива. Для по-

строения следует использовать предоставленные графические заготовки и при необходимости – встроенный графический редактор MS Excel.

- 4) *Задание 1с* предполагает построение схемы, стоящей перед входами двоичного триггера; имея два входа и два выхода, схема не позволяет появиться на выходе (и, соответственно, на входе триггера) комбинации 11. Задача имеет несколько вариантов решения – достаточно представить любой.
- 5) В *Задании 2а* необходимо «подавать» различные комбинации сигналов на входы «черного ящика» и отслеживать сигналы на выходе (комбинации «0» и «1» вводятся в ячейки, обозначенные x_1 и x_2). По этим наблюдениям заполняется таблица значений. По таблице следует установить функции выходов и по ним построить схему.
- 6) В *Заданиях 2b* и *2с* требуется по заданной таблице (функции) построить две недостающие формы представления комбинационной схемы.
- 7) По завершении работы сохраните отчет; предъявите его преподавателю.

Контрольные вопросы:

1. Что называется «дискретным устройством по обработке информации»? Почему оно называется «дискретным»?
2. Прокомментируйте классификацию дискретных устройств по обработке информации.
3. Какие дискретные устройства по обработке информации называются «комбинационными схемами»?
4. В чем особенность двоичных комбинационных схем?
5. Что такое «базис комбинационных схем»? Какие элементы он включает? Почему?
6. Каким образом можно описать комбинационную схему?
7. Укажите общий порядок решения задач анализа комбинационных схем.
8. Укажите общий порядок решения задач синтеза комбинационных схем.

Часть 2. Конечные автоматы

В базис *конечных автоматов* помимо логических вентилей входят также *элементы памяти*. К наиболее распространенным элементам памяти следует отнести *задержку*, *триггер* и *двоичный счетчик*. При выполнении заданий данной лабораторной работы будет использоваться элемент задержки, который может включаться между выходом и входом одного вентиля (образуя, тем самым, обратную цепь).

Конечный автомат описывается совокупностью *пяти* компонентов: алфавитом входных сигналов X , алфавитом выходных сигналов Y , алфавитом внутренних состояний Q , *функцией* (системой функций) *выходов*, связывающих выходные сигналы с входными сигналами и внутренними состояниями (памятью) $y_j = \theta_j(x_1, x_2, \dots, x_N, q_1, \dots, q_K)$, и функцией (системой функций) *переходов* Ψ , связывающих внутренние состояния на данном такте с выходными сигналами и внутренними состояниями на предыдущем такте $q_i = \Psi_i(x_1, x_2, \dots, x_N, q'_1, \dots, q'_K)$. Поскольку рассматриваются только двоичные устройства, входной и выходной алфавиты, а также алфавит внутренних состояний совпадают – это двоичный алфавит, содержащий знаки $\{0, 1\}$. Таким образом, двоичный конечный автомат задается логическими функциями (системами функций) переходов Ψ и выходов θ . Количество функций переходов равно числу внутренних состояний автомата (т.е. числу элементов памяти); количество функций выходов равно числу выходов схемы.

Способами описания конечных автоматов являются: схема автомата, функциональная таблица (таблица значений функций переходов и выходов), таблица истинности, диаграмма Мура. По любой из указанных форм могут быть построены остальные.

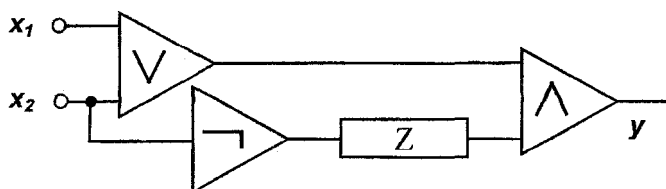
В отношении конечных автоматов выделяется две группы задач: *анализа* и *синтеза*.

Задача анализа: по имеющейся схеме автомата определить функции обработки (функции переходов и выходов), построить таблицу значений и диаграмму автомата.

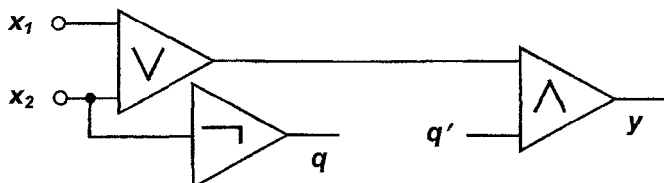
Задачи синтеза: по заданным функциям обработки, таблице значений или диаграмме построить схему автомата.

Пример решения задачи **анализа** конечного автомата.

Пусть имеется схема конечного автомата.



Для решения задачи анализа используется *метод устранения задержек* – задержка исключается из схемы, но добавляется новая вершина (q) на входе задержки и новый полюс (q') на ее выходе:



Для схемы без задержек можно записать автоматные функции:

$$y = (x_1 \vee x_2) \wedge q';$$

$$q = \bar{x}_2$$

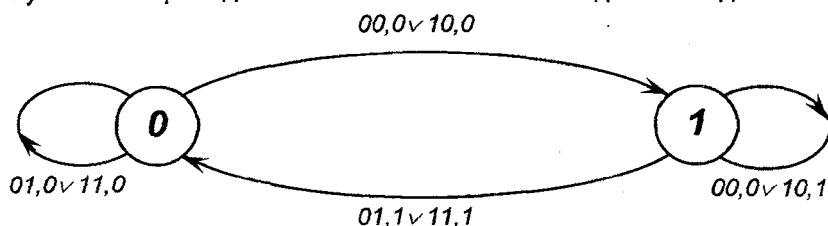
Теперь можно построить таблицу значений (таблицу истинности) для этой системы уравнений, считая все три полюса независимыми и задавая на них различные значения входных сигналов; очевидно, возможны 8 разных их сочетаний:

x_1	x_2	q'	q	y
0	0	0	1	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

На основании таблицы значений можно построить функциональную таблицу (таблицу функций). Ее строки и столбцы соответствуют входным сигналам x_i и состояниям на предыдущем такте q' . В клетках размещаются состояния текущего такта q и значение выходного сигнала y (разделяются запятой). Поскольку схема имеет два входа, в колонке x должны быть записаны все возможные сочетания входных сигналов (очевидно, их 4); при записи указывается сначала сигнал на входе x_1 , затем на x_2 .

$x \backslash q'$	0	1
00	1,0	1,0
01	0,0	0,1
10	1,0	1,1
11	0,0	0,1

На основании функциональной таблицы строится диаграмма Мура. Вершинами графа являются возможные внутренние состояния (в рассматриваемом примере 0 и 1), ребра указывают возможность перехода из одного состояния в другое; ребру приписываются условия перехода: значения сигналов на входе и выходе.



Решение задачи **синтеза** автомата из логических элементов и элементов задержки производится в обратном порядке:

- заполняется таблица значений функций автомата (по принципу: «при условии подачи на вход таких-то комбинаций на выходе должно быть то-то»); другой вариант – представляются автоматные функции в виде таблицы или диаграммы;
- по таблице значений (или таблице функций) строится система булевских функций, описывающих работу автомата;
- по функциям определяется набор логических элементов и связей между ними;
- строится схема без задержек с промежуточными полюсами и вершинами;
- вводятся элементы задержки, устраняются промежуточные полюса и вершины.

Пример решения задачи **синтеза** конечного автомата.

Пусть имеется таблица значений для некоторого конечного автомата. По ней необходимо построить функциональную таблицу и диаграмму, установить явный вид автоматных функций, а также построить схему автомата.

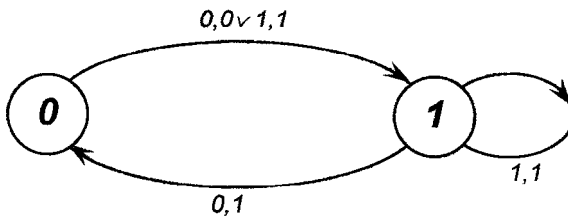
x	q'	q	y_2
0	0	1	0
0	1	0	1
1	0	1	1
1	1	1	1

Анализ условия: из таблицы видно, что схема имеет один вход (x), один выход (y) и один элемент задержки (q). Пока независимыми полюсами схемы следует считать x и q' , а вершинами – y и q . На заключительном этапе между q' и q будет включена задержка.

По таблице значений строится таблица автоматных функций:

$x \backslash q$	0	1
0	1,0	0,1
1	1,1	1,1

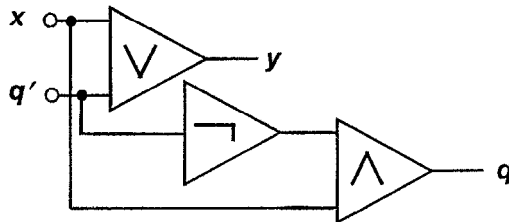
Диаграмма:



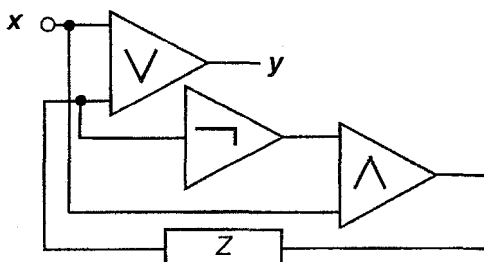
Система автоматных функций будет выглядеть следующим образом:

$$y = x \vee q'; \quad q = x \vee \bar{q}'$$

Схема без задержек с промежуточным полюсом (q') и вершиной q :



После подключения между q' и q элемента задержки окончательная схема автомата будет иметь вид:



Определить выходное слово, если входным словом (последовательность сигналов на входе) было 11001, а начальным состоянием автомата было $q = 0$.

Задача решается при использовании таблицы значений, таблицы функций или диаграммы. При этом работу автомата удобно представить по тактам:

	Такты					
Вход (x)	0	1	2	3	4	5
	—	1	1	0	0	1
Состояние (q)	0	→ 1	→ 1	→ 0	→ 1	→ 1
Выход (y)	—	1	1	1	0	1

Порядок выполнения работы:

- 1) Откройте из своей рабочей папки отчет по ЛР 7. Если требуется, установите **средний** уровень безопасности макросов. Перейдите на лист <Конечные автоматы>.
- 2) В *Задании 1* нужно методом устранения задержек решить задачу анализа для представленной схемы. При построении схем и диаграмм можно использовать имеющийся в отчете конструктор схем, а также встроенный графический редактор MS Excel. На заключительном этапе работы следует определить последовательность сигналов на выходе (выходное слово) по последовательности на входе при том что начальным является состояние автомата 00.

- 3) В *Задании 2a* необходимо «подавать» различные комбинации сигналов на входы «черного ящика» и отслеживать сигналы на выходе (комбинации «0» и «1» вводятся в ячейки, обозначенные x_1 , x_2 и q). Передача сигналов осуществляется после нажатия экранной кнопки **[Такт]**. Для очистки всех входов перед вводом значений необходимо нажать **[Сброс]**. На основании этих наблюдений заполняется таблица значений. По таблице следует установить автоматные функции, построить таблицу функций, диаграмму и схему.
- 4) В *Задании 2b* требуется по заданным автоматным функциям построить таблицу значений, таблицу функций, диаграмму, схему, а также определить выходное слово по заданному входному.
- 5) По завершении работы сохраните отчет; предъявите его преподавателю.

Контрольные вопросы:

1. Почему конечные автоматы называются «конечными»? Чем они отличаются от комбинационных схем?
2. Что представляют собой элементы памяти конечных автоматов, каковы их функции? Укажите несколько типов элементов памяти.
3. Что такое «базис конечных автоматов»? Какие элементы он включает? Почему?
4. Укажите методы описания конечных автоматов.
5. Сформулируйте правила построения диаграмм Мура.
6. В чем суть метода устранения задержек при решении задач анализа конечного автомата?
7. Укажите общий порядок решения задач анализа конечных автоматов.
8. Укажите общий порядок решения задач синтеза конечных автоматов.
9. Каким образом в конечном автомате можно определить выходные сигналы по последовательности входных?
10. В чем суть «экспериментирования» в *Заданиях 2a*?

Лабораторная работа 8.

Алгоритмическая машина Тьюринга

|| Учебная задача:
Научиться программировать эмулятор машины Тьюринга. ||

Краткие теоретические сведения

Машина Тьюринга – абстрактный исполнитель (абстрактная устройство по обработке дискретной информации). Машина Тьюринга (МТ) является расширением конечного автомата и, согласно тезису Чёрча-Тьюринга, способна имитировать все другие исполнители (с помощью задания правил перехода), каким-либо образом реализующие процесс пошаговой обработки, в котором каждый шаг обработки достаточно элементарен.

В состав МТ входит бесконечная в обе стороны лента, разделенная на ячейки (внешняя память), управляющее устройство (УУ) с конечным числом состояний (внутренняя память) и считывающе-записывающее устройство (*обозреватель*), которое на каждом шаге обозревает какую-то ячейку ленты и по команде управляющего устройства может изменить ее содержание.

МТ для конкретной задачи задается перечислением элементов множества букв алфавита $\{A\}$, множества внутренних состояний $\{Q\}$ и набором правил, по которым работает машина. Они имеют вид: $q_i a_j \rightarrow q_i' a_j' D_k$ т.е. после обзора символа a_j обозревателем при нахождении УУ в состоянии q_i , в ячейку записывается символ a_j' , УУ переходит в состояние q_i' , а лента совершает движение D_k . Для каждой комбинации $q_i a_j$ имеется *ровно одно* правило преобразования. Это означает, что УУ реализует функцию, сопоставляющую каждой паре входных сигналов $q_i a_j$ одну и только одну тройку выходных $q_i' a_j' D_k$ – она называется *логической функцией машины* и обычно представляется в виде таблицы (*функциональной схемой машины*), столбцы которой обозначаются символами состояний, а строки – знаками внешнего алфавита.

Помимо знаков внешнего алфавита $\{A\}$ на ленте может размещаться *пустой знак* – его отсылка в занятую ячейку эквивалентна стиранию ее содержания. Все ячейки, незанятые знаками $\{A\}$, считаются занятыми пустыми знаками.

В алфавит внутренних состояний УУ $\{Q\}$ помимо *рабочих состояний* (находясь в них МТ производит обработку информации) обязательно входит состояние *останова*, по достижении которого МТ прекращает работу.

Таким образом, если $\{A\}$ содержит N знаков, а $\{Q\}$ – M состояний, то общее число правил преобразования (число ячеек в функциональной таблице), очевидно, равно $(N+1) \times (M+1)$. Программирование МТ состоит в разработке данной таблицы.

Перед началом работы МТ находится в *исходной конфигурации* – в ячейки ленты занесены знаки внешнего алфавита, обозреватель располагается над некоторой ячейкой.

Работа МТ происходит тактами. На каждом такте на вход УУ подается обозреваемый в данный момент знак ленты и текущее внутреннее состояние – на основании этих данных в соответствии с таблицей преобразований выбирается команда, в соответствии с которой в ячейку записывается новый символ, машина переходит в новое состояние, а лента может быть сдвинута относительно головки на 1 ячейку вправо или влево.

В зависимости от начальной конфигурации и таблицы преобразований возможны два варианта развития событий:

- после конечного числа тактов машина останавливается по команде *останова*; при этом на ленте оказывается конечная конфигурация, соответствующая выходной информации;
- остановки не происходит (*защелкивание*).

В первом случае говорят, что данная машина применима к начальной информации, во втором – нет. На практике это означает, что были допущены ошибки при составлении таблицы преобразований.

Вся совокупность входных конфигураций, при которых машина обеспечивает получение результата, образуют

класс решаемых задач. Для каждого класса задач необходима своя МТ.

Как уже отмечалось, МТ является *абстрактной* машиной, осуществляющей обработку дискретной информации. Это означает, что построить такую машину в принципе невозможно (хотя бы потому, что элементом МТ является бесконечная лента). Вместе с тем, программно можно реализовать экранный *эмулятор* МТ, который будет обладать всеми качествами МТ, за исключением неограниченной внешней памяти. Программирование таких эмуляторов весьма полезно для освоения алгоритмизации.

В настоящей лабораторной работе предполагается использование эмулятора МТ ALGO2000 (автор Р. Зартдинов). Ниже показано размещение элементов МТ на экране:



В программе принят следующий порядок ввода информации:

- внешний алфавит вводится с клавиатуры в отведенную строку;
- начальная конфигурация записывается на ленте с клавиатуры; при попытке ввода знака, не входящего во внешний алфавит, будет выдано предупреждение; начальное поло-

жение обозревателя устанавливается нажатием экранных клавиш перемещения ленты;

- для ввода команды в ячейку нужно выделить ее щелчком мыши, после чего ввести команду с клавиатуры; приняты следующие обозначения:
 - «пробел» или знак «_» (подчеркивание) – пустой знак;
 - знаки внешнего алфавита – с клавиатуры;
 - знаки «>» и «<» – сдвиг обозревателя относительно ленты, соответственно, вправо или влево;
 - знак «!» – команда остановка; после нее указывается номер любого состояния;
 - состояние в команде задается только его номером.

Например, набор: $a > 2$; при этом в ячейку будет записано:

$a \rightarrow Q_2$.

- очистка элементов таблицы может осуществляться клавишей [Del] или с помощью меню, вызываемого правой клавишей мыши;
- запуск программы к исполнению производится при нажатии соответствующих пиктограмм в режиме *пошаговом* (трассировка) или *автоматическом*; правее находится пиктограмма принудительной остановки выполнения программы.

Порядок выполнения работы:

- 1) Скопируйте в свою рабочую папку программу ALGO2000. Запустите ее к исполнению. При необходимости переведите интерпретатор в режим *Машина Тьюринга*. Для знакомства с интерфейсом программы повторите решение задачи, изображенной на рисунке, приведенном выше.
- 2) Составьте программы решения перечисленных ниже задач 1-7 в указанном порядке. Далее решите задачи индивидуального варианта (номер получите у преподавателя). Каждую задачу сохраните в рабочей папке с указанным именем, добавив к нему свою фамилию. По завершении предъявите решения преподавателю.

Задачи:

1. **Инверсия** (файл **inv**)

Внешний алфавит: $\{0, 1\}$.

Начальная конфигурация: произвольная последовательность 0 и 1; обозревается левый символ.

МТ должна осуществить инверсию исходного слова, т.е. заменить все 0 на 1, а 1 на 0.

2. **Замена** (файл **zam**)

Внешний алфавит: $\{a, b, c\}$.

Начальная конфигурация: произвольная последовательность знаков внешнего алфавита; обозревается левый символ.

Произвести замену знаков a на b , b на c .

3. **Плюс_1** (файл **plus1**)

Внешний алфавит: $\{0, 1, +\}$

Начальная конфигурация: двоичное целое число $+1$ (например, $1101+1$); обозревается правый знак.

Необходимо осуществить суммирование произвольного двоичного числа и 1 по правилам двоичной арифметики.

4. **Плюс_1м** (файл **plus1m**)

Необходимо осуществить суммирование произвольного числа в четверичной системе счисления и 1 по правилам четверичной арифметики.

5. **Плюс_Ун** (файл **plus_u**)

Внешний алфавит: $\{1, +\}$

Начальная конфигурация: произвольные слагаемые в унарной системе счисления, разделенные знаком $+$ (например, $11111+11$); обозревается правый знак.

Необходимо осуществить суммирование чисел в унарной системе счисления.

6. **Минус_1** (файл **minus1**)

Необходимо реализовать вычитание 1 из целого числа в двоичной системе счисления.

7. **Без_a** (файл **bez_a**)

Входной алфавит $A = \{a, b, c\}$. Удалить из исходного слова все вхождения символа a .

Вариант 1.

1. $A = \{a, b\}$. В исходном слове поменять местами его первый и последний символы.
2. Необходимо реализовать умножение на 2 в унарной системе счисления.
3. Осуществить сложение двух чисел в двоичной системе отсчета с полной записью (например, $1001+11=1100$).

Вариант 2.

1. $A = \{a, b\}$. Заменить в исходном слове каждое вхождение a на bb .
2. Необходимо изменить на обратный порядок следования цифр в записи целого двоичного числа (например, $10110 \rightarrow 01110$).
3. Осуществить сложение произвольного троичного чисел с одноразрядным троичным числом с полной записью (например, $1012+2=1021$).

Вариант 3.

1. $A = \{a, b, c\}$. Заменить в исходном слове каждое вхождение ab на c .
2. Необходимо реализовать умножение на 4 целого двоичного числа.
3. Осуществить вычитание двух чисел в двоичной системе отсчета с полной записью (например, $1001-11=110$).

Вариант 4.

1. $A = \{a, b\}$. Удвоить слово исходное слово (например: $abb \rightarrow abbabbb$).
2. Необходимо преобразовать унарную запись числа в двоичную.
3. Осуществить вычитание одноразрядного числа из произвольного в троичной системе счисления с полной записью (например, $1021-2=1012$).

Вариант 5.

1. $A = \{a, b\}$. Удвоить каждый символ исходного слова (например: $bab \rightarrow bbaabb$).
2. Необходимо реализовать умножение на 9 целого троичного числа.

3. На ленте запись двоичного числа. Необходимо определить, является ли оно записью степени двойки (1, 2, 4, 8, ...) в двоичной системе счисления. Ответ: 1 (является) или 0.

Вариант 6.

1. $A = \{a, b\}$. Перевернуть исходное слово (например: $abb \rightarrow bba$).

2. Необходимо у двоичного числа удалить незначащие нули, если такие есть (изначально обозревается произвольная цифра числа).

3. На ленте запись числа в четверичной системе счисления. Необходимо определить, является ли оно четным. Ответ: 1 (является) или 0.

Контрольные вопросы:

1. Почему машину Тьюринга можно считать предельным случаем конечного автомата? Почему МТ не является конечным автоматом?
2. Почему машина Тьюринга называется абстрактной?
3. Укажите назначение основных компонентов МТ?
4. Что такое «конфигурация МТ»?
5. Какова структура команды в МТ? В чем суть программирования МТ?
6. Опишите функционирование МТ в процессе решения задачи.
7. Если в процессе исполнения алгоритма на МТ происходит «зацикливание», о чем это может свидетельствовать?
8. Можно ли построить универсальную МТ, способную решить любую задачу по обработке информации в дискретной форме представления?
9. Что общего и чем различаются МТ и компьютер? Можно ли считать компьютер машиной Тьюринга?
10. Почему программа, которая была использована в данной лабораторной работе, называется «эмулятор МТ»?

Библиография

1. Пильщиков В.Н. и др. Машина Тьюринга и алгоритмы Маркова. Решение задач. – М.: МГУ, 2006. – 47 с.
2. Стариченко Б.Е. Теоретические основы информатики: Учебное пособие для вузов. – М.: Горячая линия – Телеком, 2003. – 312 с.
3. Стариченко Б.Е. Теоретические основы информатики: Электронный учебный курс. Екатеринбург, УрГПУ, 2006. <http://elib.uspu.ru>
4. Фомин С.В. Системы счисления. – М.: Наука, 1987. – 48 с.

Учебное издание

**Стариченко Борис Евгеньевич
Горнов Олег Александрович**

**Лабораторный практикум
по курсу «Теоретические основы информатики»**

Оригинал-макет подготовлен Б.Е. Стариченко

ЛР № 040330 от 18.04.97

Подписано в печать 20.01.2009 Формат 60х84 ¹/₁₆
Бумага для множ. апп. Печать на ризографе.
Гарнитура Таймс. Усл. печ. л. 3,4. Тираж 200 экз.
Заказ 2702.

Тираж отпечатан в отделе множительной техники
Уральского государственного педагогического университета
620017, г. Екатеринбург, пр. Космонавтов, 26
E-mail: uspu@uspu.ru